International Journal of Computers, Communications & Control
Vol. I (2006), No. 2, pp. 7-14

Development of a Matlab® Toolbox for the Design of Grey-Box Neural Models

Gonzalo Acuña, Erika Pinto

Abstract: A Matlab Toolbox is developed for the design, construction and validation of grey-box neural network models. This toolbox, available in www.diinf.usach.cl/gacuna has been tested in simulations with a continuously stirred reactor process. The grey-box model performs well for validation data with 5% additive gaussian noise for one-step-ahead (OSA) and model-predictive-output (MPO) estimations.
Keywords: Grey-Box Model, Neural Networks, One-Step-Ahead estimation, Multiple Prediction Output, Time variant parameter identification.

1 Introduction

In the development of dynamic system models it is best to take advantage of the a priori knowledge of a process, generally expressed in terms of a set of ordinary differential equations which represent mass or energy balances. In complex biotechnological processes, the most difficult task is the modeling of the time-varying parameters, such as specific kinetics. In order to address this problem [1], proposed the use of grey-box models which combine a priori knowledge expressed in terms of a phenomenological, or white-box model, with a black-box model such as neural network. These models have proved to be satisfactory for dynamic systems, they have better generalization characteristics, and they can be identified with a smaller amount of data [1]. [2] classified these grey-box models into two principal categories: those which deliver intermediate values (of parameters or variables) for use in phenomenological models (serial grey-box models), or those in parallel with the dynamic model, adjusted to compensate for modeling errors (parallel grey-box models). [3] showed that the series strategy resulted in grey-box models with superior results. More recently [4] and [5] have employed and analyzed this type of model demonstrating their performance and their use in complex processes.

Matlab® is well known as a completely integrated application development environment, oriented to projects that involve complex mathematical calculations and graphic visualizations. This software has a large variety of toolboxes which are specialized packets which can carry out different functions according to the area of development, for example; optimization, image processing, neural networks, simulations and statistics among others. Thus, the present work deals with the creation of a group of functions integrated in a Matlab® Toolbox which allows the development of an Grey-Box Neural Model (GBNM) for complex systems in general.

The present document is organized as follows: Section 2 details some of the aspects of the GBNMs used, Section 3 examines the design and construction of the Toolbox; Section 4 looks at the applications of the Toolbox; and Section 5 outlines the conclusions reached.

2 Grey-Box Neuronal Models

As mentioned previously GBNMs take advantage of the combination of the a priori knowledge of a given process -expressed in terms of a set of differential equations that represent the first principles that govern the process- with neural networks. The latter are responsible for modeling the interaction between variables that are relevant to the system, and certain time-varying. It is a well established fact that neural networks are capable of approximating non-linear functions. In particular, it has been demonstrated that perceptrons, with only one hidden layer and an adequate number of neurons in their internal layer, are

universal approximators [6].

For the purposes of the present work it is important to distinguish between two training modes for neural networks included in GBNMs. The first type, also known as the direct learning mode [5], uses the error generated at the output of the neural network for the correct determination of its weights (Figure 1).

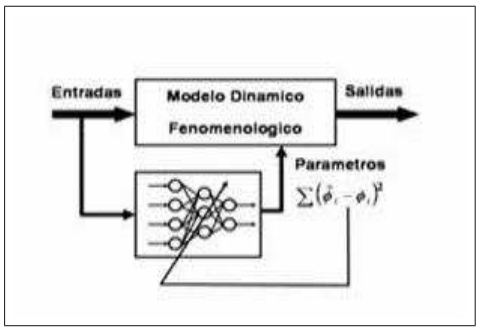

Figure 1: Grey-Box Neural Model in its direct learning mode.

The second type corresponds to an indirect mode by which the error generated at the output of the GBNM is used for the training of the neural network [5] (Figure 2).

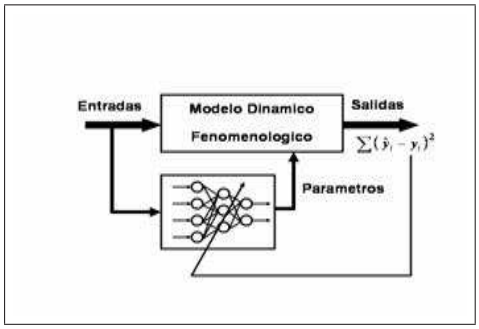

Figure 2: Grey-Box Neural Model in its indirect learning mode.

The Toolbox developed in the present work is based on the direct learning mode of the neural network. The neural networks used are multi-layered perceptrons with only one hidden layer. The training algorithm is error retropropagation combined with a Levenberg-Marquardt optimization.

The validation of the results obtained is carried out with tests that consist in evaluating the error produced when using the GBNM for a One Step Ahead (OSA) prediction, and for Model Predictive Output (MPO) prediction, [7]. The error indices used are three: the Root Mean Square (RMS) error, the Relative Standard Deviation (RSD) and the Adecuation Index (IA), which are presented below [7]:

$$RMS = \sqrt{\frac{\sum_{i=1}^{n}(o_i - p_i)^2}{\sum_{i=1}^{n} o_i^2}}; \qquad RSD = \sqrt{\frac{\sum_{i=1}^{n}(o_i - p_i)^2}{N}}; \qquad IA = 1 - \frac{\sum_{i=1}^{n}(o_i - p_i)^2}{\sum_{i=1}^{n}(|o_i'| + |p_i'|)^2}$$

Where o_i and p_i are the observed and predicted values respectively, in time i, and N is the total number of data. $p_i' = p_i - o_m$ and $o_i' = o_i - o_m$, where o_m is the median value of the observations.

3 Development of the Matlab Toolbox

This Toolbox consists of two principal parts: a Neural Network and serial Grey Box Model, either with OSA or MPO estimates.

The methodology used corresponds to the Modified Cascade Model which has the following stages:

- Definition of requirements.
- System Design.
- Implementation and testing of units.
- Integration and testing of system.

The details of the methodology can be found in [8].

3.1 Neural Network

This is the first part of the Toolbox, and its principal function is to create and train the network that the serial Grey-Box Model uses for estimating the unknown parameter or parameters which are then combined with the phenomenological part of the model. For this reason it is necessary to run this part of the program before running the serial Grey-Box Model.
The principal functions of the neural network are the following:

- Verify the existence of data: This function's role is to verify the existence of the data necessary for the functioning of the neural network.
- Verify dimensions: This function verifies that the quantity of input output data is equal.
- Verify layers: This function verifies that the total number of neurons in hidden layers is correct.
- Divide data: This function divides input and output data that have been entered by the user into training and validation data, (70% and 30% respectively).
- Create and train the network: This function creates and trains the neural retropropagation network based on the data input by the user. This is carry out using the Matlab® neural network Toolbox functions. For the creation of the neural network the "trainlm" function is used which applies the Levenberg-Marquardt method for optimizing weights.
- Simulate network outputs: This function simulates network outputs for the purpose of validation.
- Graph network inputs: This allows the visualization of the behavior of the entered input.

3.2 Serial Grey Box Model

This is the second part of the Toolbox which allows the development of general serial Grey-Box Models either with OSA or MPO estimations.

The principal functions of the Serial Grey-Box Model applied either to OSA or MPO estimations are the following:

- Verify the existence of the files: This function verifies the existence of the files necessary for the functioning of the serial Grey-Box Model.
- Verify the existence of data: Among these data are the number of iterations, the initial values for the status vector, the value of delta t, and where applicable to input u and the real values of the status vector variables for each instant of time used.

- Verify dimensions: This function verifies that the dimensions of the input vector be the same as the number of iterations. For estimating OSA the quantity of real values of the status variables must be equal to the number of iterations. For MPO it only returns the results for the output of the system, but cannot evaluate said output with real data.
- Develop Model: This function is in charge of developing the Grey-Box Model which consists of estimating the unknown parameter or parameters by using the neural network created and trained previously in order to develop the different equations entered.
- Return different graphs: Graphs the inputs of the system, each of the components of the status vector for the total number of iterations, the validation of the unknown parameter or parameters, and the output of the system together with its validations.
- Calculate the error indices: Calculates RSD, RMS and IA for the output.
- Display results: returns the last iteration both for the status vector and the output vector, and for the error indices that correspond to the unknown parameter or parameters or the output or outputs of the system.

4 Aplications and Results

4.1 Description of the process

In the simulation of a CSTR process a first order exothermic reaction is used [9], in which the inputs to the system correspond to the temperature of the cooling sleeve and the system output corresponds to the degree of completion of the reaction. The systems is described by the following status equations:

$$x_1' = -x_1 - D_a \cdot (1-x_1) \cdot e^{\left(\frac{x_2}{1+\frac{x_2}{\gamma}}\right)} \tag{4.1}$$

$$x_2' = -x_2 - B.D_a \cdot (1-x_1) \cdot e^{\left(\frac{x_2}{1+\frac{x_2}{\gamma}}\right)} + \beta.(u-x_2) \tag{4.2}$$

$$y = x_1 \tag{4.3}$$

Where: x_1 : Degree of completion of the reaction; x_2 : Adimensional temperature of the reactor contents; u : Input that corresponds to the adimensional flow rate of the heat-transfer fluid through the cooling sleeve, and y is the output of the system.

For the purposes of this simulation the following values are used for the model's constants:

$$D_a = 0,072; \qquad B = 8,0; \qquad \beta = 0,3; \qquad \gamma = 20,0$$

Because Grey-Box Models are made up of two parts, one of which is the phenomenological model represented by the different differential equations, and the other of which corresponds to the empirical model represented by the neural network, the equations that represent the phenomenological model of the serial Grey-Box Model are the following:

$$x_1' = -x_1 + 0,072.(1-x_1).\rho \tag{4.4}$$

$$x_2' = -x_2 + 8.0,072.(1-x_1).\rho + 0,3.(u-x_2) \tag{4.5}$$

$$y = x_1 \tag{4.6}$$

Where ρ is the parameter that is difficult to obtain:

$$\rho = e^{\left(\frac{x_2}{1+\frac{x_2}{20}}\right)}$$

When equations 4.4 through 4.6 are discretized in the interval between t and t+1, the following equations are obtained:

$$x_{1(t+1)} = x_{1(t)} + (-x_{1(t)} + 0,072.(1-x_{1(t)}).\rho).\Delta t \tag{4.7}$$

$$x_{2(t+1)} = x_{2(t)} + (-x_{2(t)} + 8.0,072.(1-x_{1(t)}).\rho + 0,3.(u-x_{2(t)})).\Delta t \tag{4.8}$$

Thus the equations that constitute the phenomenological part of the serial Grey-Box Model, either with OSA or MPO estimations, are the following:

$$x_{1(t+1)} = x_{1(t)} + (-x_{1(t)} + 0,072.(1-x_{1(t)}).\rho).\Delta t \tag{4.9}$$

$$x_{2(t+1)} = x_{2t} + (-x_{2(t)} + 8.0,072.(1-x_{1(t)}).\rho + 0,3.(u-x_{2(t)})).\Delta t \tag{4.10}$$

$$y = x_{1(t+1)} \tag{4.11}$$

4.2 OSA Estimation with 5% noise

The validation obtained for the unknown parameter with 5% noise can be seen in Figure 3.

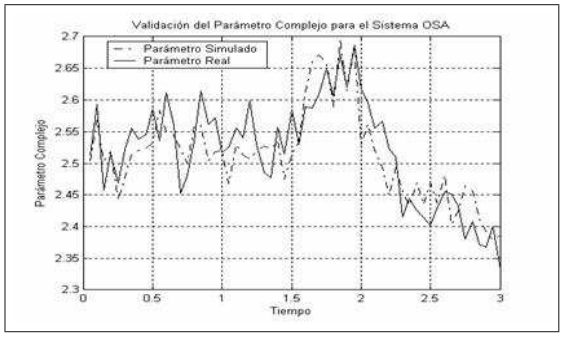

Figure 3: Validation of the unknown parameter with OSA estimation and 5% noise.

The validation obtained from the output of the system can be seen in Figure 4.

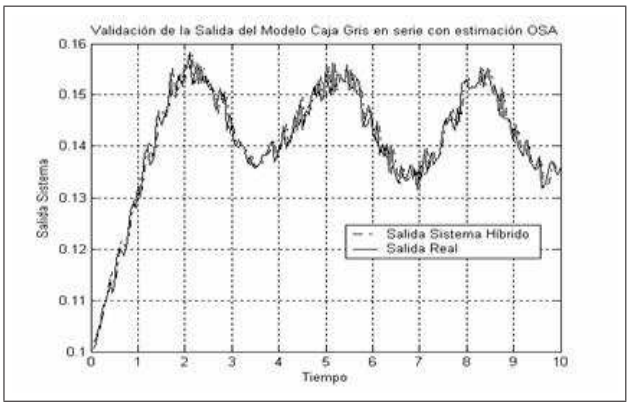

Figure 4: Validation of the output of the system with OSA estimation and 5% noise.

The error indices obtained from the serial Grey-Box Model with OSA estimation and 5% noise, for parameter ρ and the output of the system are presented in Table 4.1.

Table 4.1. Error indices for parameter and the output of the system with an OSA estimation and 5% noise.

	Parameter ρ	System Output
RSD	4.368098E-2	2.756748E-3
RMS	1.733075E-2	1.944067E-2
IA	9.111975E-1	9.846065E-1

The error indices obtained are quite acceptable given that the acceptability values are $RSD < 0.1$, $RMS < 0,1$ and $IA > 0.9$, as explained in Section 2.6 This indicates that the serial Grey-Box Model developed for the simulation of a CSTR process, with OSA estimation and 5% noise, is quite good.

4.3 MPO Estimation with 5% noise

The validation obtained for the unknown parameter with 5% noise can be seen in Figure 5.

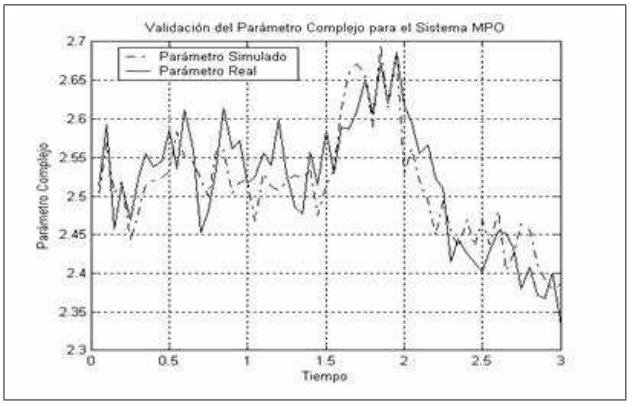

Figure 5: Validation of the unknown parameter with MPO estimation and 5% noise.

The validation obtained from the output of the system can be seen in Figure 6.

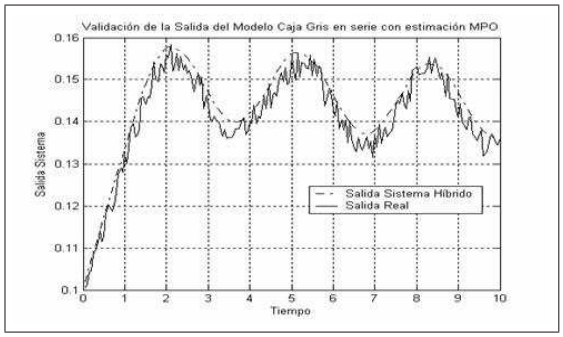

Figure 6: Validation of the output of the system with MPO estimation and 5% noise.

The error indices obtained from the serial Grey-Box Model with MPO estimation and 5% noise, for parameter ρ and the output of the system are presented in Table 4.2.

Table 4.2. Error indices for parameter ρ and the output of the system with an MPO estimation and 5% noise.

	Parameter ρ	System Output
RSD	4.368098E-2	3.441815E-3
RMS	1.733075E-2	2.427178E-2
IA	9.111975E-1	9.764686E-1

The error indices obtained are quite acceptable given that the acceptability values are $RSD < 0.1$, $RMS < 0,1$ and $IA > 0.9$. This indicates that the serial Grey-Box Model developed for the simulation of a CSTR process, with MPO estimation and 5% noise, is quite good.

5 Conclusions

Grey-Box Models constitute a real alternative for those real world processes for which the available a priori knowledge is incomplete, for example in a variety of industrial processes. As in Grey-Box Models only some of the physical and/or chemical laws that represent the model are known, and there are unknown parameters that must somehow be estimated, multi-layered perceptron neural networks have been employed for their notable capacity to approximate complex functions on the basis of observed data.

One of the advantages of this Toolbox, with reference to the creation and training of neural networks, is that neuron transfer functions for each of the hidden and output layers that make up the neural network, can be determined by the user, who can then create and train the neural network in a manner that is in keeping with the specific problem that is to be modeled.

Concerning the estimation of unknown system parameters, the developed Toolbox has the capacity to estimate as many parameters as necessary as long as the necessary data are available.

The developed Toolbox (available at www.diinf.usach.cl/gacuna), allows the creation of serial Grey-Box Models, either with OSA (One Step Ahead) or MPO (Model Predictive Output) estimations, depending on what is to be modeled. With the creation of this Matlab® Toolbox, the end user has access to a simple, trustworthy and fast tool which allows the development and subsequent manipulation of different serial Grey-Box Models, either with OSA or MPO estimations for solving a particular problem. This tool will become part of the Matlab® Toolboxes such that the user can access it at will simply by downloading Matlab®. Thus this new tool becomes a real and simple alternative for modeling those real work processes for which a priori knowledge available is incomplete.

The error indices obtained are good, both for the unknown parameters and for each of the outputs of the system, even when the level of noise applied to the various input data was 5%, which indicates that the models developed are of good quality.

Acknowledgements

The authors wish to thank partial financing provided by Fondecyt Project 1040208.

References

[1] Psichogios, D.; Ungar, L. (1992). “A Hybrid Neural Network-First Principles Approach to Process Modeling”, Computers & Chemical Engineering, 38(10): 1499-1511.

[2] Thompson, M.; Kramer, M. (1994). “Modeling Chemical Processes Using Prior Knowledge and Neural Networks”, Computers & Chemical Engineering , 40(8):1328-1340.

[3] Van Can, H; Hellinga, C.; Luyben, K.; Heijnen, J. (1996). “Strategy for Dynamic Process Modeling Based on Neural Network in Macroscopic Balances”, AIChE Journal, 42:3403-3418.

[4] Thibault, J, Acuña, G., Pérez-Correa, R., Jorquera, H., Molin, P., Agosin, E., (2000) “A hybrid representation approach for modelling complex dynamic bioprocesses” Bioprocess Engineering, 22(6):547-556.

[5] Acuña G.; Cubillos, F.; Thibault, J.; Latrille, E. (1999). “Comparison of Methods for Training Grey-Box Neural Network Models”, Computers & Chemicals Engineering Supplement, 23:561-564.

[6] Hornik K. Stinchcombe M., White H., (1989) “Multilayer Feedforward Networks Are Universal Approximators”, Neural Networks, 2:359-366.

[7] Billings S.A., Jamaluddin H. B. y Chen S., (1992) “Properties of Neural Network with Applications to Modeling Non-linear Dynamical System”, Int. J. Control, 55(1):193-224.

[8] Pinto Armijo, Erika (2004), “A Matlab based application for developing grey-box models”, Memoria de Título de Ingeniería Civil Informática, Universidad de Santiago de Chile (in Spanish).

[9] Hernández E.; Arkun, Y. (1992). “Study of the Control-Relevant Properties of Backpropagation Neural Network Models of Nonlinear Dynamical Systems”, Computers & Chemical Engineering, 16(4):227-240.

Gonzalo Acuña, Erika Pinto
Universidad de Santiago de Chile
Departamento de Ingeniería Informática
Avda. Ecuador No 3659 - Casilla 10233; Santiago, Chile
E-mail: gacuna@usach.cl